Introduction à la Cryptographie

5. Cryptographie asymétrique (1)

Cécile Pierrot, Chargée de Recherche INRIA Nancy
cecile.pierrot@inria.fr

Supports de E. Thomé

Telecom Nancy, 2A ISS – 2021

Plan

Mise en œuvre de crypto symétrique

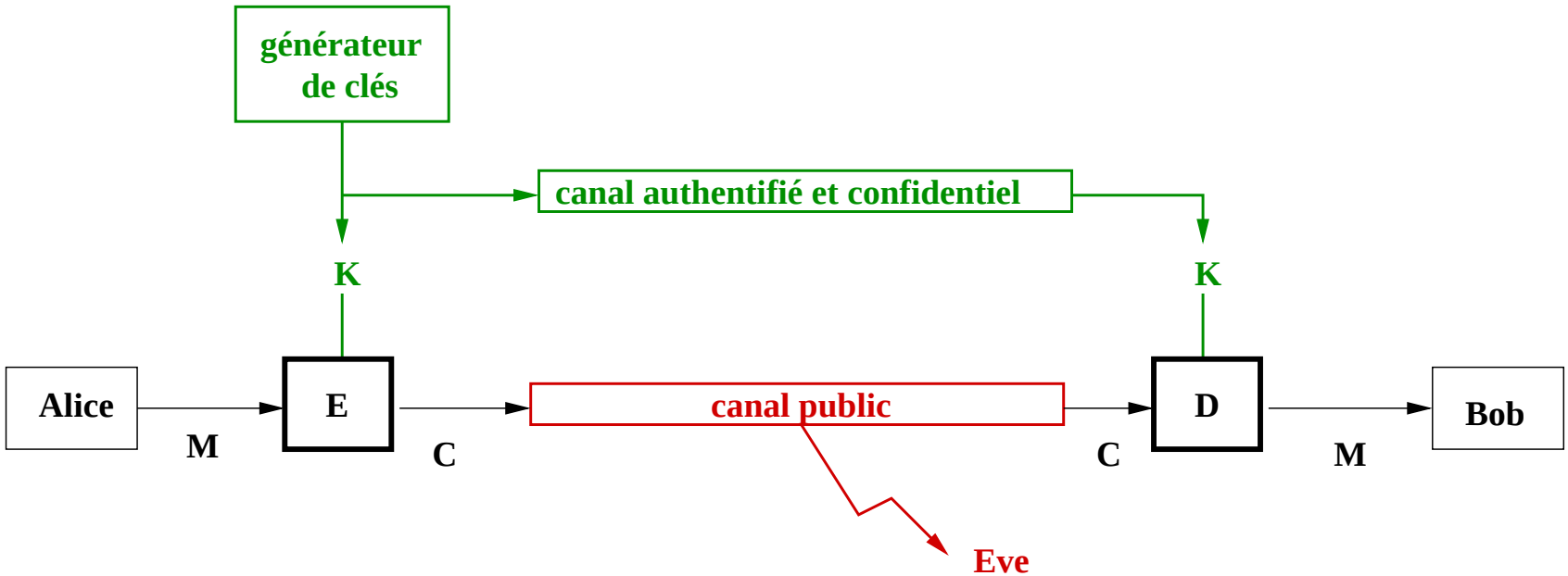
Clé publique

RSA

Propriétés de sécurité

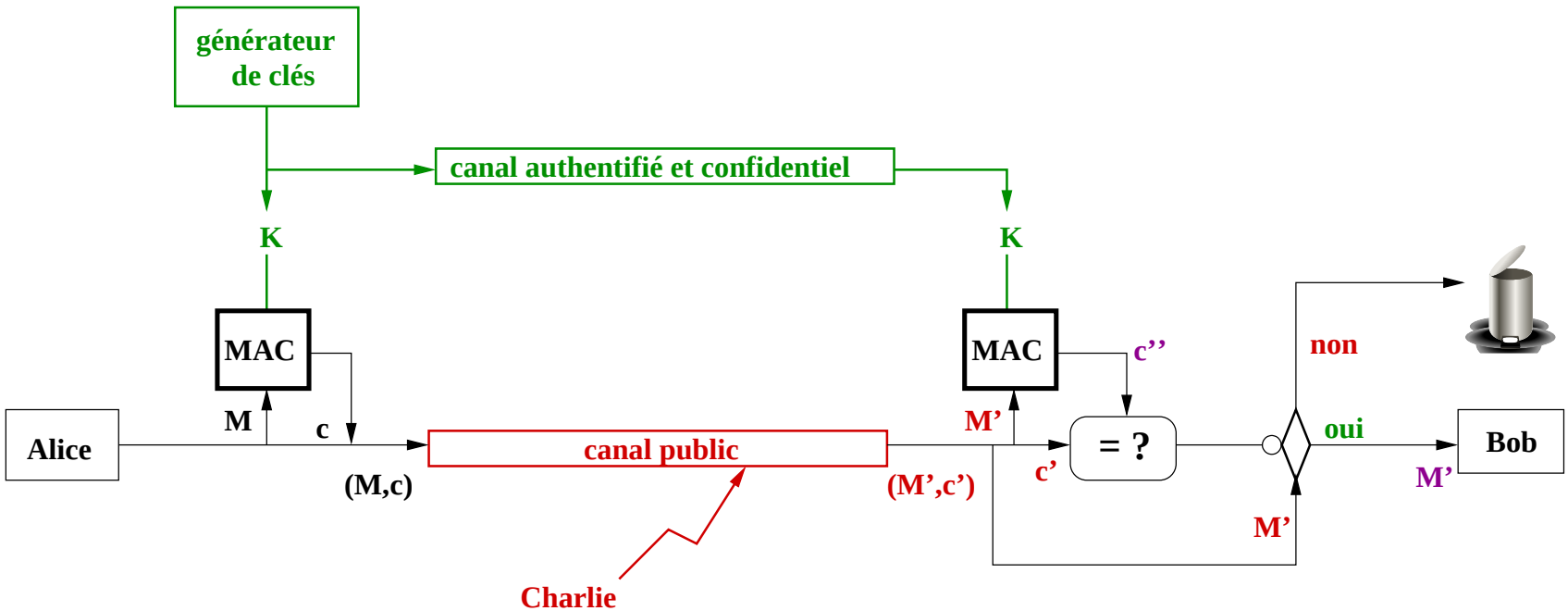
Difficulté de la factorisation

Créer un canal confidentiel



- On a besoin du canal authentifié et confidentiel au préalable de l'envoi d'un nombre élevé de messages

Créer un canal authentifié



- On utilise le canal authentifié et confidentiel préalablement au message

Observations

Deux défauts majeurs du contexte symétrique :

- Difficulté de l'échange de clés initial.
- Symétrie : si A et B connaissent K, il peuvent faire la même chose avec.
  - Pas possible de signer
  - Une clé par couple (A, B). n^2 clés pour n participants.

Plan

Mise en œuvre de crypto symétrique

Clé publique

RSA

Propriétés de sécurité

Difficulté de la factorisation

Crypto à clé publique

La cryptographie à clé publique rend possible le contexte où :

- Alice dispose d'une clé de chiffrement k_A, publique.
- Bob dispose d'une clé de déchiffrement k_B, secrète.

Cette asymétrie rend notamment possible la signature.

Outils

La cryptographie à clé publique repose sur la difficulté de problèmes mathématiques.

- En connaissant k_A, il est possible de chiffrer.
- En connaissant k_B, il est possible de déchiffrer.
- Le problème difficile est : découvrir k_B à partir de k_A :
  - calculatoirement hors de portée
  - (mais possible en un temps infini).
- Autre problèmes, pas forcément équivalents (peut-être plus faciles) :
  - Déchiffrer un message chiffré avec k_A.
  - Distinguer le chiffré de oui et le chiffré de non.

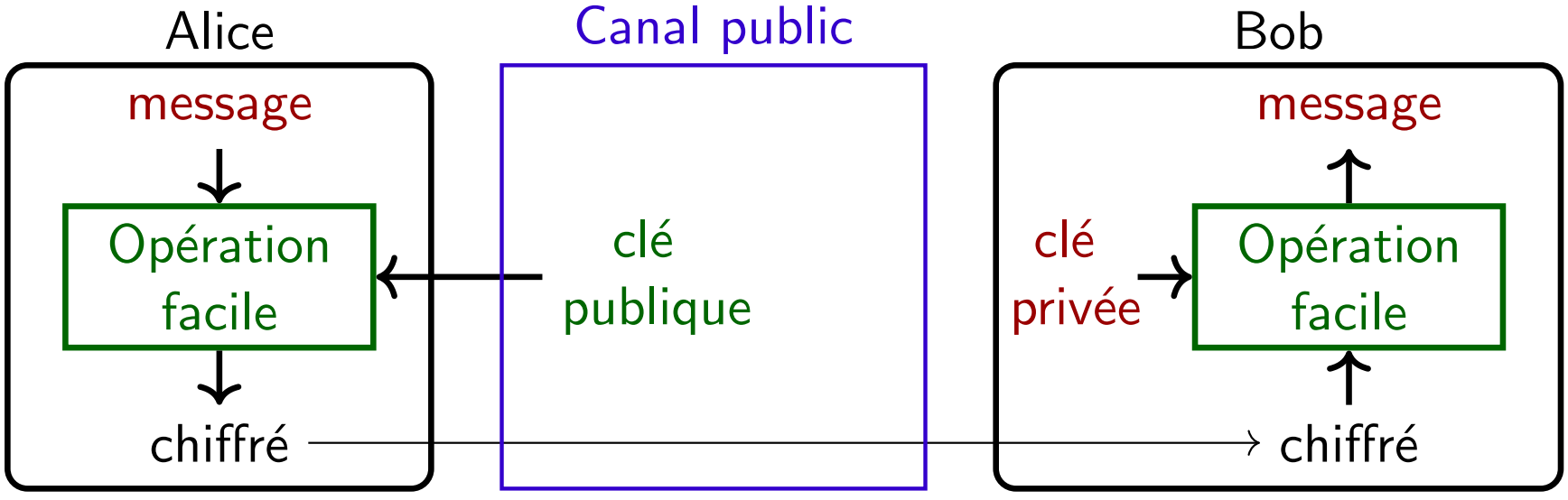
Outils

La cryptographie à clé publique repose sur la difficulté de problèmes mathématiques.

- En connaissant k_A, il est possible de chiffrer.
- En connaissant k_B, il est possible de déchiffrer.
- Le problème difficile est : découvrir k_B à partir de k_A :
  - calculatoirement hors de portée
  - (mais possible en un temps infini).
- Autre problèmes, pas forcément équivalents (peut-être plus faciles) :
  - Déchiffrer un message chiffré avec k_A.
  - Distinguer le chiffré de oui et le chiffré de non.
    (le chiffrement doit être non-déterministe)

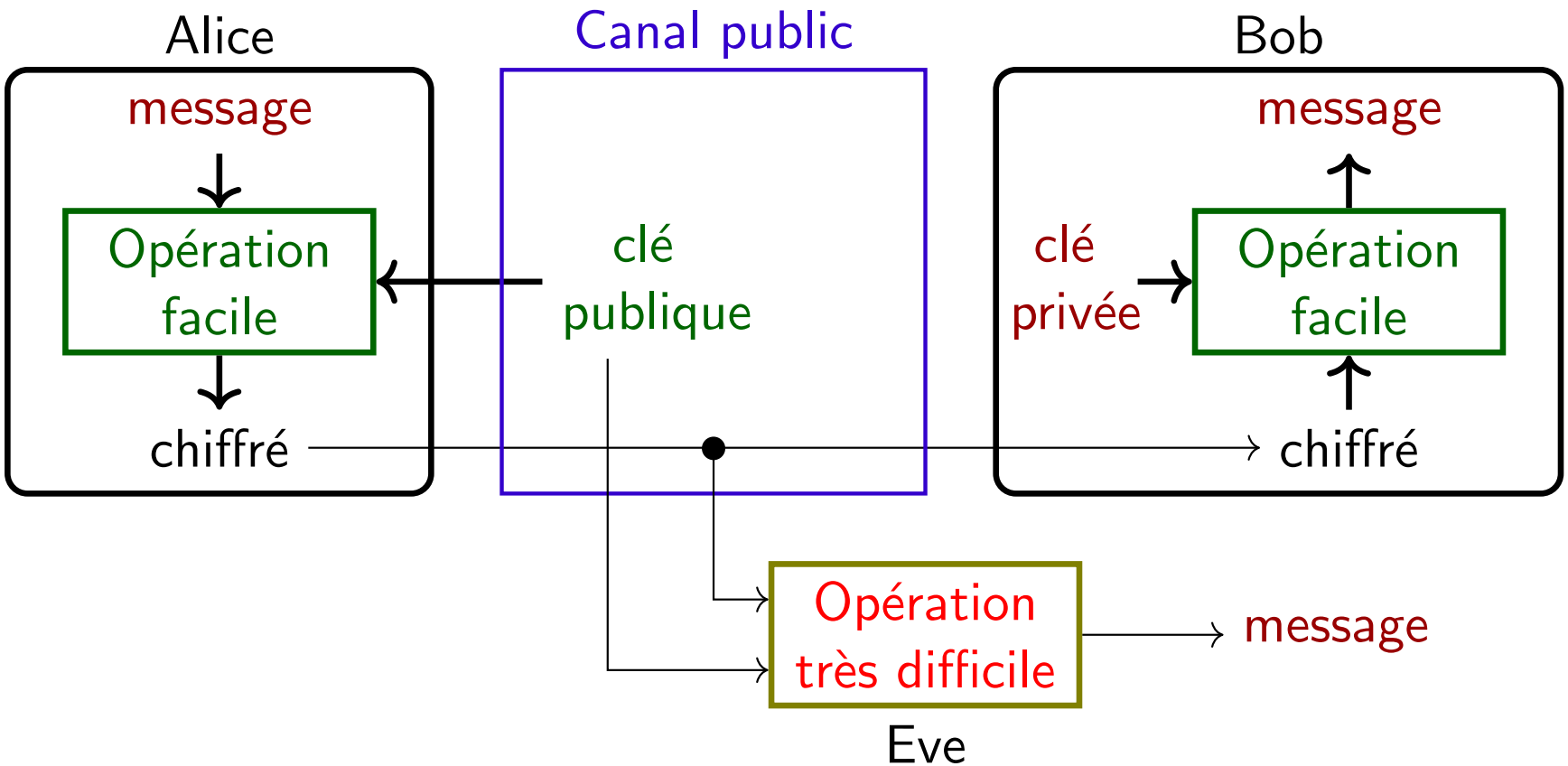
Reposer sur un problème mathématique (1)

Alice envoie un message chiffré à Bob.



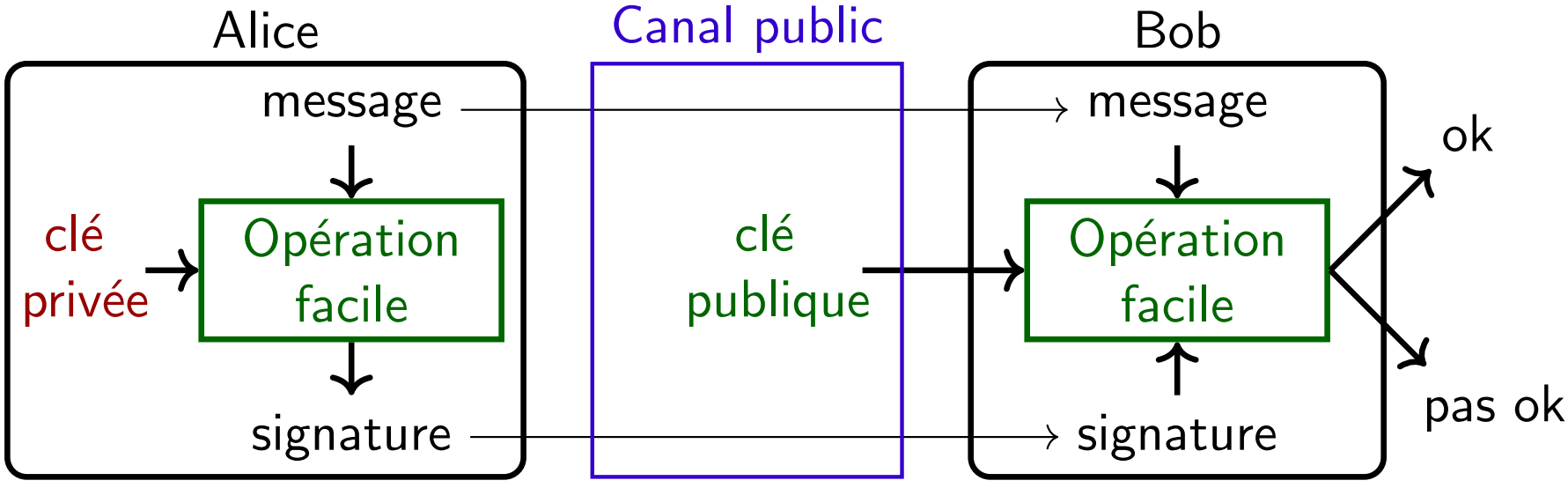
Reposer sur un problème mathématique (1)

Alice envoie un message chiffré à Bob.



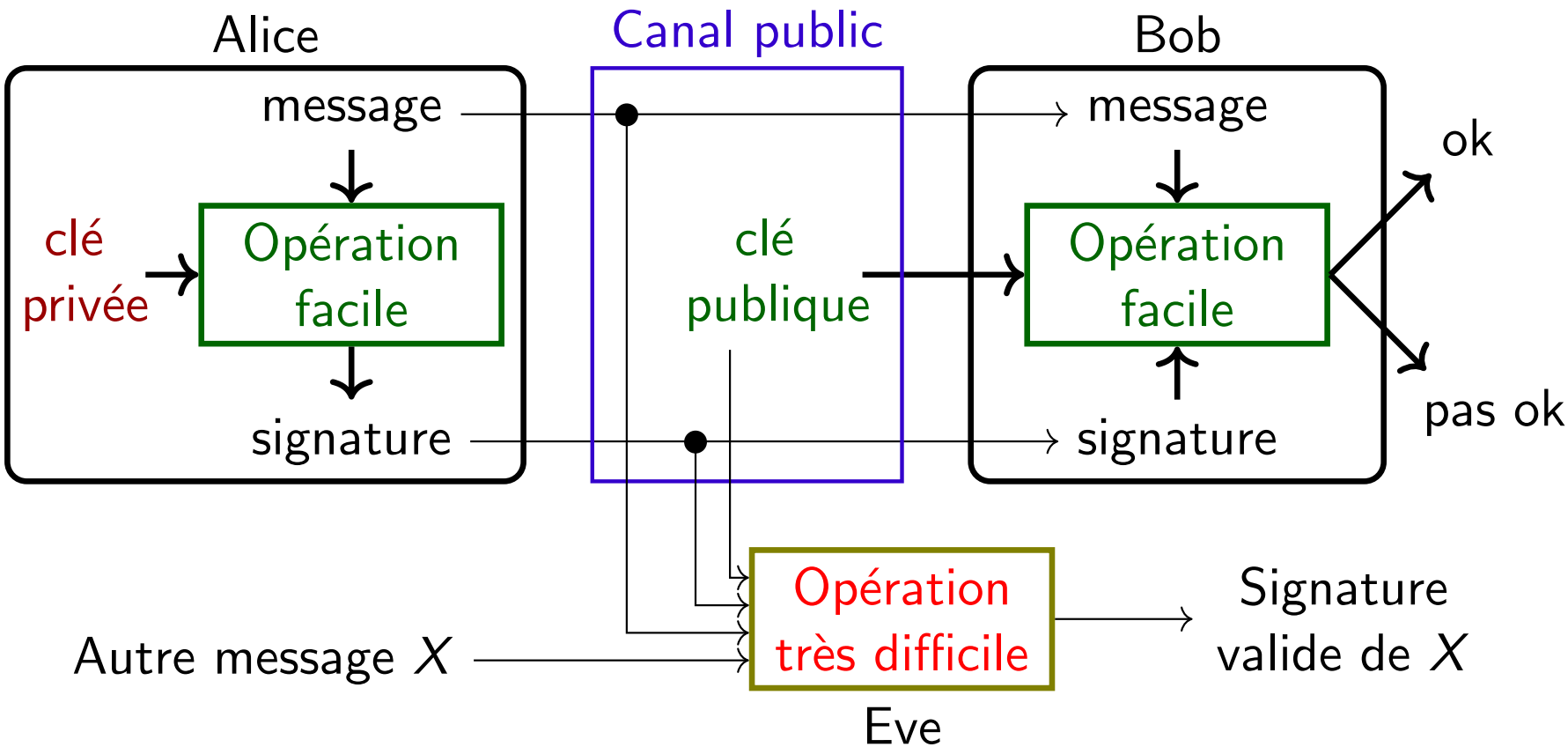
Reposer sur un problème mathématique (2)

Bob vérifie un message signé par Alice.



Reposer sur un problème mathématique (2)

Bob vérifie un message signé par Alice.



Outils

Les problèmes mathématiques couramment rencontrés en crypto.

Pour les plus fréquents :

- Problème de la factorisation d'entiers (cryptosystème RSA).
- Problème du calcul de logarithme discret dans les groups (échange de clés).

Mais aussi :

- Problèmes liés à la théorie des codes correcteurs.
- Recherche de vecteurs courts dans des réseaux de $\mathbb{Z}^n$.
- etc, etc.

S'appuyer sur un pb mathématique = ?

L'idéal : prouver que si E ≪casse≫ le cryptosystème $\mathcal{S}$, alors il sait résoudre un problème très difficile H.

Rêve extrême : si E ≪casse≫ $\mathcal{S}$, alors il sait résoudre toute instance d'un problème NP-complet H donné (pour une taille fixée). Alors :

- Soit $P = NP$ (tant mieux pour lui).
- Soit on a montré que le E a du pain sur la planche.

S'appuyer sur un pb mathématique = ?

L'idéal : prouver que si E «casse» le cryptosystème $\mathcal{S}$, alors il sait résoudre un problème très difficile H.

Rêve extrême : si E «casse» $\mathcal{S}$, alors il sait résoudre toute instance d'un problème NP-complet H donné (pour une taille fixée). Alors :

- Soit $P = NP$ (tant mieux pour lui).
- Soit on a montré que le E a du pain sur la planche.

C'est beau de rêver, mais on n'a pas ça. On a :

- Si E sait résoudre le problème H, il casse $\mathcal{S}$,
- H est une instance d'un problème difficile.
- En l'état actuel des connaissances, on ne sait résoudre H qu'en s'attaquant au problème difficile.

Systèmes étudiés

On présente :

- Le cryptosystème RSA.
- Le protocole de Diffie-Hellman.

Plan

Mise en œuvre de crypto symétrique

Clé publique

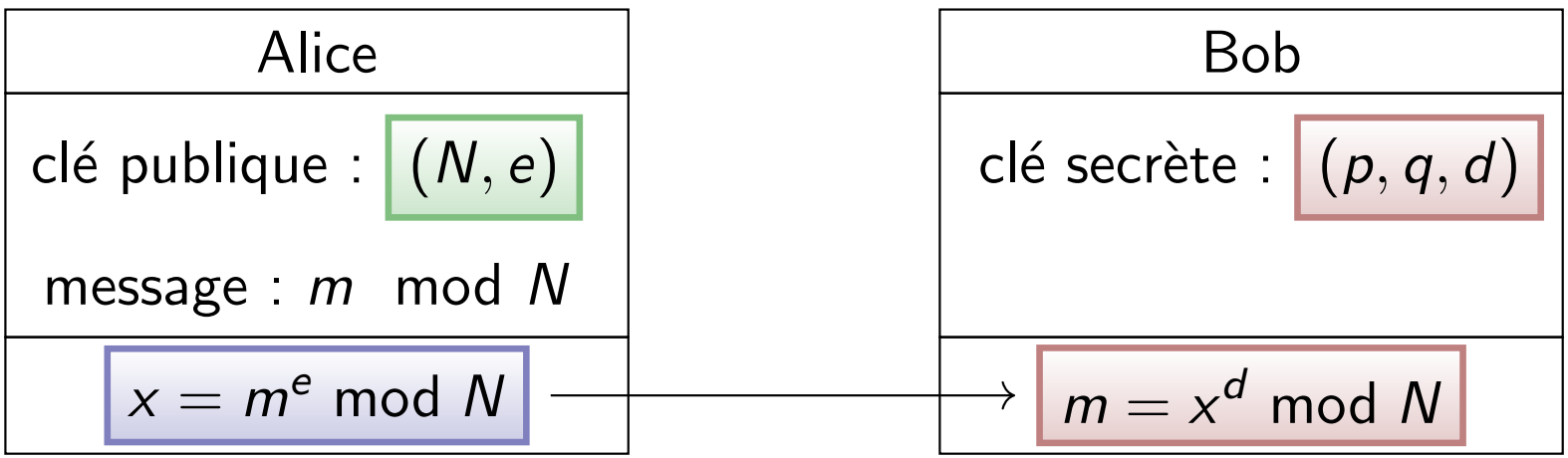
RSA

Propriétés de sécurité

Difficulté de la factorisation

Le chiffrement RSA

- N entier, et p, q (premiers) tels que $N = pq$.
- $\phi(N) = (p-1)(q-1)$
- e premier avec $\phi(N)$. L'entier e est inversible modulo $\phi(N)$.
- d calculé tel que $de \equiv 1 \mod \phi(N)$.



$$x^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \mod N.$$

RSA : exemple

En TD !

Signature RSA

Pour signer un message m :

- Bob calcule $s = m^d \mod N$.
- Alice vérifie que $s^e \equiv m \mod N$.

Preuve de RSA

On a $ed \equiv 1 \mod \phi(N)$, et $\phi(N) = (p-1)(q-1)$.

En particulier, on a $ed = 1 + k(p-1)(q-1)$, donc $m^{ed} = m^{1+(p-1)\times \text{cst.}}$.

- Petit théorème de Fermat : $m^{p-1} \equiv 1 \mod p$.
  Donc $m^{ed} \equiv m \mod p$.
- Si jamais $p \mid m$, alors $p \mid m^{ed}$, donc $m^{ed} \equiv m \mod p$.

Idem pour q, donc $pq \mid (m^{ed} - m)$, $\square$.

Que peut faire un espion ?

- S'il sait factoriser N, il a gagné.
- S'il ne peut pas, est-il impossible de calculer $\sqrt[e]{x \bmod N}$?
- Factoriser N est la meilleure approche connue pour $\sqrt[e]{}$.
- Hypothèse RSA : $\sqrt[e]{x \bmod N}$ est difficile si N assez grand.

Le choix des clés pour RSA

- Un module RSA est un nombre composé N.
- Un module RSA ne sert qu'à une seule personne.

Il faut définir une procédure pour générer des clés.

- Choisir un nombre premier p aléatoire de $n/2$ bits.
- Choisir un nombre premier q aléatoire de $n/2$ bits.
- Choisir e, calculer d tel que $de \equiv 1 \mod (p-1)(q-1)$.
- Publier N, conserver p, q.

Le choix du nombre de bits n est crucial.

Juste une primitive

La fonction RSA $c = m^e \bmod N$ n'est pas un cryptosystème, c'est une primitive. Reste à détailler :

- Quelques propriétés désagréables.
- Principes de mise en œuvre.
- Choix des clés.

Propriétés désagréables de RSA

Tautologie : sous l'hypothèse RSA, la fonction RSA possède la propriété ≪ sens unique ≫.

Étant donné un chiffré, il est difficile de retrouver le clair associé.

La primitive RSA a la particularité d'être un morphisme :

$$m_1^e m_2^e \equiv (m_1 m_2)^e \mod N.$$

- On peut arguer que $m_1 m_2$ n'est pas un message valide.
- Néanmoins inacceptable, surtout si des preuves sont requises.

En outre : un message m est associé à un unique chiffré c.

- Facile de distinguer le chiffré de oui et le chiffré de non.
- Le résultat du chiffrement doit être variable ;

Cf plus loin dans le contexte du ≪ padding ≫.

Plan

Mise en œuvre de crypto symétrique

Clé publique

RSA

Propriétés de sécurité

Difficulté de la factorisation

Objectifs possibles pour l'attaquant

Étant donné un système asymétrique et une paire k_A, k_B.
Différents objectifs pour l'attaquant :

- Cassage total : retrouver k_B à partir de k_A.
- Inversion : déchiffrer (une partie d') un message chiffré avec k_A (avec bonne probabilité).
- Attaque sémantique : distinguer le chiffré de oui et le chiffré de non (avec proba $> \frac{1}{2} + \epsilon$).

Moyens possibles pour l'attaquant

Principe général des lois de Kerckhoffs : l'attaquant est fort, il n'a juste pas la clé (secrète).

Plusieurs contextes d'attaque. Soit E l'attaquant.

- Attaques à chiffrés seuls : E connaît une liste de chiffrés.
- Attaques à clairs connus : E connait une liste de paires (m_i, c_i).
- Attaque à clairs choisis : E peut obtenir le chiffré c_i correspondant à des messages qu'il choisit.

Moyens possibles pour l'attaquant

Principe général des lois de Kerckhoffs : l'attaquant est fort, il n'a juste pas la clé (secrète).

Plusieurs contextes d'attaque. Soit E l'attaquant.

- Quelques capacités ≪built-in≫ liées à la clé publique :
  - Attaques à chiffrés seuls : E connaît une liste de chiffrés.
  - Attaques à clairs connus : E connait une liste de paires (m_i, c_i).
  - Attaque à clairs choisis : E peut obtenir le chiffré c_i correspondant à des messages qu'il choisit.
- Scénarios plus ambitieux :
  - Attaque à chiffrés choisis : E peut demander le déchiffrement d'un certain nombre de chiffrés c_i.
  - Attaque à chiffrés choisis adaptative : E peut ajuster ses requêtes en fonction des réponses, et de son objectif.

Terminologie barbare

On donne des petits noms aux objectifs et aux scénarios d'attaque.

Objectifs :

- OW (one-wayness) : le cryptosytème résiste à l'inversion.
- IND (indifférentiabilité) : résiste à l'attaque sémantique.

Attaques :

- CPA : Chosen Plaintext Attack.
- CCA : Chosen Ciphertext Attack.
- CCA2 : Adaptative Chosen Ciphertext Attack.

Un cryptosystème est :

- OW-CPA si l'inversion résiste à une attaque CPA.
- IND-CCA2 si l'indifférentiabilité résiste à une attaque CCA2.

Relations :

$$x\text{-CCA2} \Rightarrow x\text{-CCA} \Rightarrow x\text{-CPA},$$

$$\text{IND-}y \Rightarrow \text{OW-}y.$$

OW-CPA = pas assez

Tautologie : la fonction RSA possède la propriété « sens unique ».

i.e. la primitive RSA est « OW-CPA ».

On veut à la place du IND-CCA2 (plus fort !).

C'est l'objet du « padding ».

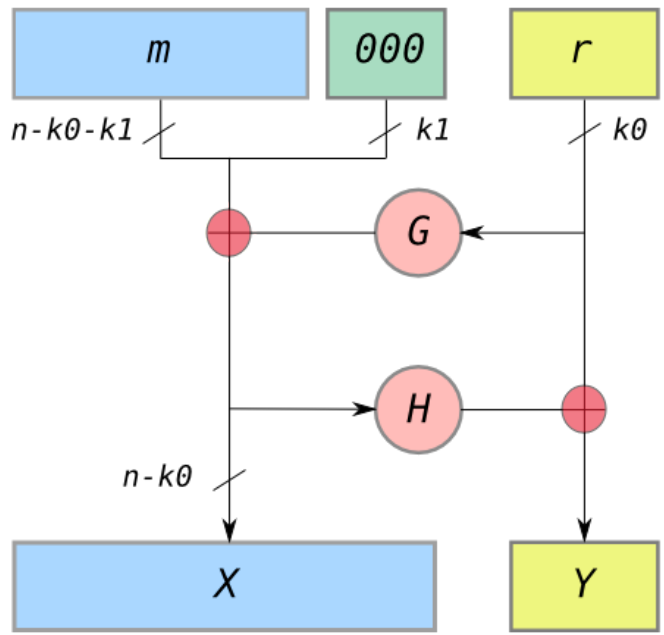
Détails de mise en œuvre

Le schéma de padding : composant essentiel du chiffrement RSA.

- Point de départ : primitive qui fait peu de choses.
- À l'arrivée : fonction de chiffrement avec des propriétés fortes.

Schéma de padding OAEP :

$$X \parallel Y = \mathrm{OAEP}(m, r),$$
$$c = f(X \parallel Y).$$



- Les fonctions de hachage jouent ici un rôle capital.
- Normalisé dans des standards. Effectivement utilisé.

Plan

Mise en œuvre de crypto symétrique

Clé publique

RSA

Propriétés de sécurité

Difficulté de la factorisation

Choix des clés

Hypothèse RSA : $\sqrt[e]{x \bmod N}$ est difficile si N assez grand.

- On n'est même pas sûr que c'est aussi dur que de factoriser N.
- Il faut au moins choisir N assez grand pour qu'il soit difficile de le factoriser.

Si on veut de la sécurité dans la durée, il faut avoir une idée de la difficulté de la factorisation dans le futur.

Record de factorization : N de 829 bits, en 2020. Environ 2700 ans CPU !

Algorithmes pour factoriser N

Pour factoriser un entier N de n bits.

- Divisions successives : $O(\sqrt{N}) = O(2^{n/2})$ $[-\infty]$
- Méthode ρ : $O(\sqrt[4]{N}) = O(2^{n/4})$ [1975]
- CFRAC : $\exp\left(O(\sqrt{n})\right)$ [1975]
- Crible Quadratique $\exp\left(O(\sqrt{n})\right)$ [1982]
- Crible Algébrique $\exp\left(O(\sqrt[3]{n})\right)$ [1990]

Difficulté de la factorisation

- 1999 : factorisation 512 bits.
- 2010 : factorisation 768 bits.
- 2020 : factorisation 829 bits.

Avec le crible algébrique, (NFS) la taille de clé n qu'il faut pour que l'effort de l'attaquant soit égal à 2^s est :

$$n = O(s^3).$$

- La puissance de calcul croît $\pm$ exponentiellement...
- ...donc s croît linéairement...
- ...donc la taille de clé pour RSA monte, monte, monte.

Aujourd'hui : l'ANSSI ne valide pas l'usage de clés RSA de moins de 2048 bits dans les produits crypto (3072 si usage après 2030).

Il existe encore des clefs de 1024 bits... correspond à 2^{77} opérations, à éviter !